UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/729,773 | 12/08/2003 | Jeffrey B. Scott | I20 04984US | 3422 |

128 7590 04/15/2009

HONEYWELL INTERNATIONAL INC.
101 COLUMBIA ROAD
P O BOX 2245
MORRISTOWN, NJ 07962-2245

| EXAMINER | |
|---|---|
| WANG, JUE S | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2193 | |

| MAIL DATE | DELIVERY MODE |
|---|---|
| 04/15/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/729,773 | SCOTT, JEFFREY B. | |
| | Examiner | Art Unit | |
| | JUE S. WANG | 2193 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *17 December 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-7 and 9-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-7 and 9-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1-7 and 9-12 have been examined.

2. Claim 8 was cancelled in Amendment dated 7/31/2008.

### *Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-7, 9, 10, and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Defaix et al. (US 2004/0133444 A1, hereinafter Defaix), view of Hammack et al. (US 6,449,624 B1, hereinafter Hammack).

5. As per claim 1, Defaix teaches a source control system comprising:

a processor in a source control system (see Fig 1, [0027]);

an import function operable on said processor in response to a request by a user to import an object from an external source (see [0035]; EN: import function is performed when the user at a client wants to place a new revision of an object into the repository);

a validation function operable on said processor to determine whether said import object is eligible for automatic check-in (see [0035]; EN: the validation function is performed when checking whether the client is allowed to check in the new version), wherein said validation function comprises:

determining if said import object already exists as an existing object in said process control system (see [0032], [0035]);

if said import object already exists as said existing object, determining if said existing object has a status of checked-in (see [0007], [0035]; EN: determining if a file is locked where the filed is locked through a locked checkout);

if said existing object is checked in, determining if said user has permission to check-in (see [0039], [0040], claim 15, part c; EN: using a access control list to validate requests by the client); and

a check-in function operable on said processor to be performed automatically upon import if said import object is validated by said validation function, including determining a version number for said import object (see [0029], [0035]).

Defaix does not explicitly teach locking said status of said existing object so as to validate said import object. However, Defaix teaches a locked checkout mechanism such that only the developer who owns the lock can modify the file by checking in a new version (see [0029]). It would have been obvious to one of ordinary skill in the art at the time of the invention to lock the status of the existing object to prevent other developers from modifying the file when the file is being checked in.

Defaix does not teach that the source control system is for a process control system with a processor.

Hammack teaches a process control system with a processor and version control functions for process configurations such as import and check-in (see Figs 1, 4, 6-8, abstract, column 2, lines 29-67, column 8, line 33 – column 13, line 57, column 16, lines 19-57).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the source control system of Defaix with the import function into the process control system of Hammack because it is desirable to provide version control for process configurations in a process control system since multiple process operators modifying the process configuration stored in a configuration database will lead to version control problems when one operator is unaware of the work done by another operator (see column 1, line 51 – column 2, line 15 of Hammack).

6. As per claim 2, Defaix does not teach that said object defines a control strategy.

Hammack teaches that the object tracked in the version control database defines control strategies (see Fig 3, column 6, line 19 – column 7, line 41).

7. As per claim 3, Defaix does not teach that the system comprising at least one controller capable of being loaded with said control strategy by said processor.

Hammack teaches at least one controller capable of being loaded with said control strategy by said processor (see Fig 1, item 12, column 3, lines 54-59, column 6, lines 19-24).

8. As per claim 4, Defaix teaches the system further comprises at least one client in communication with said processor (see Fig 1, [0026]).

9. As per claim 5, Defaix does not teach said control strategy is distributed from said processor to said at least one client.

Hammack teaches that the control strategy is distributed from said processor to said at least one client (see column 6, lines 19-24).

10. As per claim 6, Defaix teaches a database accessible by a processor to store said object (see Fig 1, [0026]).

11. As per claim 7, the limitations recited in this claim are substantially similar to those recited to claim 1. Therefore, it is rejected using the same reasons as claim 1.

12. As per claim 9, Defaix teaches unlocking said status of said existing object, after said import object has been automatically checked-in (see [0007], [0035]).

13. As per claim 10, Defaix teaches the check-in procedure comprises:

determining if said import object already exists as an existing object in said source control system ([0032], [0035]);

if said import object already exists as said existing object, determining if a status of said existing object is locked (see [0007], [0035]; EN: determining if a file is locked where the filed is locked through a locked checkout);

determining a new version number for a new version of said existing object (see [0029]);

checking-in said import object as said new version using said new version number (see [0029], [0035]).

Defaix does not teach storing a comment in said source control system indicating an automatic check-in for said new version.

Hammack teaches storing a comment after an import (see column 16, line 60 – column 17, line 3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Defaix to store a comment as taught by Hammack because the use of comments are well known in version control systems to provide additional information about a particular version.

14. As per claim 12, the limitations recited in this claim are substantially similar to those recited to claim 2. Therefore, it is rejected using the same reasons as claim 2.

15. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Defaix et al. (US 2004/0133444 A1, hereinafter Defaix), view of Hammack et al. (US 6,449,624 B1, hereinafter Hammack), as applied to claim 10 above, further in view of Shiman et al. (US 2002/0019827 A1, hereinafter Shiman).

16. As per claim 11, Defaix and Hammack do not teach that determining said new version number for said new version comprises: determining an existing version number of said existing object; determining an import version number from said import object; setting said new version number to a minor increment of said existing version number, if said import version number is equal to said existing version number; setting said new version number to a major increment of said existing version number, if said import

version number is less than said existing version number; and setting said new version number to said import version number, if said import version number is greater than said existing version number.

Shiman teaches a method for managing documents in a centralized document repository (see abstract), where a new version number is determined for files uploaded to the repository, comprising: determining an existing identify tag including existing version number (see Fig 4, Fig 27, step 2712, [0074], and [0196]); determining an upload version number from the uploaded file (see Fig 4, Fig 7, step 2701, [0074], and [0193]), setting the version number to a minor increment of said existing version number, if said uploaded version number is equal to said existing version number (see Fig 27, steps 2710, 2712, and [0196]). Shiman does not explicitly teach setting the new version number to a major increment of said existing version number, if said uploaded version is less than said existing version. However, Shiman teaches setting the new version number to be a major increment when the Branch tag of the uploaded file does not match an existing tag (see Fig 4, Fig 27, steps 2703, 2704, [0074], [0078], [0194]). Similarly, it would have been obvious to one of ordinary skill in the art at the time of the invention that a new branch with a major increment of the version number is created if the uploaded version number is less than the existing version number because the uploaded file must be from a new branch of development. Shiman also does not explicitly teach setting the new version number to the uploaded version number, if the uploaded version number is greater than the existing version number. However, Shiman teaches setting the new version to the next available minor version not used when the uploaded version is greater than the existing version (see Fig 27, steps 2710, 2711, [0196]). It would have been

obvious to one of ordinary skill in the art that setting the new version number to the next available minor version achieves the same result as setting the new version number to the version number of the uploaded file because the next available minor version must be greater than all currently used minor versions since minor versions are incremented by one each time the owner submits a new document (see [0078]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Defaix and Hammack to perform determining an existing version number of said existing object; determining an import version number from said import object; setting said new version number to a minor increment of said existing version number, if said import version number is equal to said existing version number; setting said new version number to a major increment of said existing version number, if said import version number is less than said existing version number; and setting said new version number to said import version number, if said import version number is greater than said existing version number as taught by Shiman because it allows the import function to import objects that already have version numbers associated and to use the version number to appropriately integrate the imported object into the source control system by constructing a new version number based on the version number of the imported object.

***Response to Arguments***

17. Rejection of claims 1 and 7 under 35 U.S.C.§103(a):

18. As per independent claims 1 and 7, Applicant argued that Defaix does not disclose any of the four steps as a part of the validation function. With respect to the first

of the four steps, Applicant argued that the files that the client 300 uses can only be files stored in repository 102, the server has no need to make a determination of whether the import object already exists in the process control system as claimed.

Applicant's arguments have been fully considered and Examiner respectfully disagrees. Examiner submits Defaix not only teaches using files stored in the repository, but also teaches files that are not initially stored in the repository, i.e., "some files will also already be stored in the sandbox", see [0032]. In addition, it would have been obvious to one of ordinary skill in the art that a repository associated with a software development environment does not start in an initial state pre-populated with a fixed set of files, instead, developers are allowed to add new files to the repository over time. Examiner further submits that because not all files are initially stored in the repository, the operation of checking whether a file is locked performed with a request to check in a file ([0035]) would necessarily check whether the file exists first since the locked status of a file can not be determined for a file that does not yet exist in the repository.

19. With respect to the second step of the validation function, Applicant argued that Defaix merely discloses checking to see if the file is locked to another client and does not check to see if the import object has a status of checked in.

Applicant's arguments have been fully considered and Examiner respectfully disagrees. Examiner submits that checking if a file is locked is the same as checking if a file is checked in since a file is locked when a file is checked out (see [0007]). Therefore, if a file is locked, then it is not checked in, and if a file is not locked, then it is checked in.

20. With respect to the third step of the validation function, Applicant argued that it is questionable that the password procedure described in paragraph 0039 and 0040 is used by the validation function described in paragraph 0035. Since client 300 can only use files stored in repository 102 of server 102 of server 100 (paragraph 0032), it must be used when client 300 requests the file for its sandbox 306 and there is no description that once this request is granted, the control list would later be consulted for a request concerning the file already checked out to client 300.

Applicant's arguments have been fully considered and Examiner respectfully disagrees. Examiner submits that that the access control list is used to check whether the client is allowed to checked in the new version since the access control list is used to control who has access to objects in the repository (see [0039]) and manage requests from clients to modify the repository (see claim 15, part c). A check in of a new version is a request to modify the repository, therefore, Examiner submits that access control is used by the validation function described in paragraph [0035].

21. With respect to the fourth step of the validation function, Applicant argued that Examiner's contention that "it would have be obvious to lock the status of the existing object to prevent other developers from modifying the file when the file is being checked in" is erroneous because the locked checkout mechanism disclosed by Defaix (0007) locks an object upon checkout and not upon check-in as recited in claimed fourth step of the validation function.

Examiner respectfully disagrees with this argument. Examiner submits that Defaix teaches that a check in is allowed when the file is not locked (see [0035],

specifically "if the file is locked, then only the owner of the lock can check in a new version"). Examiner submits that under the situation where a check in is performed on a file that is not locked, it would have been obvious to lock the file while the check in is in progress to prevent other developers from modifying the file when the file is being checked in.

***Conclusion***

22. **THIS ACTION IS MADE FINAL**. See MPEP §706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

23. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

Jue Wang
Examiner
Art Unit 2193